

**TITLE****METHOD AND ENCODER FOR IMPLEMENTING A FULLY PROTECTED  
MULTIDIMENSIONAL LINEAR BLOCK CODE****INCORPORATION BY REFERENCE OF PRIORITY DOCUMENT**

This application is based on, and claims the benefit of, the European Patent Application No. 02 292 309.8 filed on September 20, 2002, which is incorporated by reference herein.

**BACKGROUND OF THE INVENTION****1. Field Of The Invention**

The present invention relates to the field of the correction of errors that may have been generated during the transmission of a signal. In particular, it concerns a method and an encoder for implementing a Linear Block Code (LBC) characterised by a great length and a correcting capacity higher than that of the known solutions.

**2. Description Of The Prior Art**

Linear block codes are known, and their use in the field of telecommunications is also known. In order to implement the code, an encoder is provided at the transmitting end and a decoder (error corrector) at the receiving end.

The encoder receives of a certain number of strings of  $k$  information symbols (information to be transmitted) as input and supplies strings of  $n$  symbols, with  $n > k$ , as an output. The receiver receives the strings of  $n$  symbols, which may be more or less corrupted by the transmission channel, and reconstructs the  $k$  symbols of the information string.

In other words, the receiver compensates for the errors introduced by the channel on the basis of the  $n - k$  redundancy symbols added during transmission. If, for example, the symbols are bits, there are  $2^k$  possible strings of information. Each of the  $2^k$  strings of code corresponds to each string. The code is therefore these  $2^k$  words of  $n$  bits, as a whole.

The linearity consists of the fact that the combination (the sum, for example) of two code strings/words is also a code string.

As can be imagined, the correcting capacity of the code depends (as well as on its structure), on the length ( $n$ ) and on the number of redundant symbols ( $n - k$ ). Generally speaking, performance rises as ( $n - k$ ) increases and, the  $k/n$  ratio being equal, it rises as the length ( $n$ ) increases.

To complete the description, a definition of systematic code is also provided. A "systematic code" is defined as a code that allows the ( $k$ ) information symbols to pass through unaltered and adds the ( $n - k$ ) parity symbols.

The procedure for calculating the  $n - k$  parity symbols starting from the  $k$  information symbols is well known, and is a linear operation (overlapping of the effects applies).

The interleaving of two linear block codes with the aim of achieving a correcting capacity higher than that of conventional linear block codes, as in so-called Product Vector Codes (PVC's), is a known procedure. The main disadvantage of PVC's consists in the creation of the rectangular interleaver (since the code words are the rows and the columns of the structure, in order to pass from one code to another code the interleaver writes by rows and reads by columns and *vice versa*).

Block code structures with unprotected dual redundancy are also known. The disadvantage of such code structures is that they provide a lower code correcting capacity. Indeed, while the information symbols are bi-protected, the parity/redundancy symbols are not.

Finally, partially protected block code structures with double redundancy that only partly solve the problem of the lowered correcting capacity of the code are known.

### **SUMMARY OF THE INVENTION**

The main object of this invention is providing a method for carrying out error correction with a high correcting capacity.

A further object of this invention is that of providing a method for carrying out error correction using an interleaver that is simplified as compared with the known ones.

Yet another object of this invention is that of providing a method for carrying out error correction in which each parity is self-protected, thus giving rise to a completely bi-protected or even n-protected block code, in which the redundancies protect each other.

A further object of this invention is that of making the size of the code more flexible, avoiding the constraints that characterise PVC's.

A further object of this invention is that of providing a simplified processing algorithm, so as to simplify the hardware as compared with the known systems.

According to the present invention, these and other objects are achieved by means of a code with double and reciprocal protection, in which the interleaver is obtained by carrying out a permutation within columns and, possibly, a permutation between columns.

The structure can furthermore be generalised to cases in which there are more than two protections.

The method according to the invention is characterised by the steps indicated in claim 1; the encoder of this invention includes the features of claim 7. The respective dependent claims define further features that are peculiar to the invention. All the claims are understood to be integral parts of this description.

According to a preferred embodiment of the invention, in the interleaver the permutation within the columns comprises a rotation of the elements of the columns.

According to a preferred embodiment of the invention, a permutation is carried out between columns on the redundancies.

According to a preferred embodiment of the invention, the code provides bi-protection of the redundant symbols.

The length of the superblock will be given by the length of one of the codes (the two codes both have the same length  $n$ ) multiplied by an arbitrary parameter.

The present invention will certainly become clear from the following detailed description, to be read with reference to the attached figures.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

In the drawings:

- Fig. 1 shows a structure of a systematic Product Vector Code with protection;
- Fig. 2a shows the structure of a block code with double redundancy without protection of the redundancies;

- Fig. 2b shows a partially protected block code with double redundancy (interleaved redundancy A is protected by redundancy B);
- Fig. 2c shows an example of a block code with mutually protected double redundancy (redundancy A with interleaver A is protected by redundancy B and redundancy B with interleaver B is protected by redundancy A);
- Figures 3a and 3b show a possible implementation of this invention with two sets of redundancies, in which one rotation is carried out on the data and one rotation on the redundancy symbols;
- Fig. 3c, associated with Fig. 3a, shows a variation of the embodiment illustrated in Fig. 3b in which one of the possible exchanges of columns is carried out on the redundancy;
- Figures 4a-4c illustrate schematically a possible embodiment of this invention with three sets of redundancies; and
- Fig. 5 shows an encoder according to a preferred embodiment of the invention, with two sets of redundancies.

#### **BEST MODE FOR CARRYING OUT THE INVENTION**

Reference should be made initially to Fig. 1, which shows the structure of a systematic Product Vector Code. Given two codes with parameters  $(n_1, k_1)$  e  $(n_2, k_2)$ , the Product Vector Code organises the symbols into a two-dimensional table  $(n_1 \times n_2)$  in which the  $n_2$  rows are code words (having a length of  $n_1$ ) and the  $n_1$  columns are code words (having a length of  $n_2$ ).

One of the disadvantages of PVC's lies in the creation of the rectangular interleaver. Furthermore, one is forced to take the product of the two lengths ( $N = n1 \times n2$ ,  $K = k1 \times K2$ ) as the length of the superblock. In addition, in spite of these constraints, an effective supercode is in any case often not obtained.

Fig. 2a shows the structure of a block code with double redundancy without protection of the redundancies. This structure provides for calculation of two sets (A and B) of redundancy symbols calculated by rows. It is, basically, a PVC with the bottom right-hand corner for carrying out the cross check missing.

The structure of Fig. 2a is advantageous in terms of rate (since it is not necessary to transmit the "Check on Check". The basic disadvantage is that while the information symbols are bi-protected, the parity/redundancy symbols are not. It is therefore sufficient for a pattern of errors that cannot be corrected to occur on the parity (on parity A for example) to make the whole block uncorrectable.

Fig. 2b shows an example of a partially protected block code with double redundancy (the first set of symbols A is protected but the other set of symbols B is not protected). It must be noted that redundancy A is only transmitted once. The only constraint, in this case, is that  $n2$  must be smaller than  $n1$ , that is to say the number of rows must be smaller than or equal to the length of the code.

In the case of a partially protected block code with double redundancy, the problem of the fragility of the code is partly solved: the codes have different lengths, ( $n2 > n1$ ), and (the only) parity A is bi-protected. First of all parity A is calculated and then, after the interleaving by columns, parity B is calculated.

The invention consists in the interleaving of two (or more) LBC's to obtain a super-LBC that is far longer than the two starting codes, and therefore results in a high correcting

capacity. The present invention is proposed as an alternative to the PVC, and its main features are a lower complexity and the non-existence of particular constraints in terms of length and rate, which are typical, on the other hand, of a PVC. This invention also has the advantage that it can be extended to a multidimensional code rather than being limited to bi-dimensional codes.

According to this invention, the construction of the code is obtained by means of two consecutive operations, i.e. a first operation for generating the interleaver and a second operation for constructing the self-protected parities of the code. For the sake of simplicity and clarity, the two operations will be described separately.

Interleaver:

Generally speaking, the object of any interleaver is to combine the elements in such a way that the number of elements that the codes have in common is as small as possible.

Fig. 3a shows a block of data/information *Inf* to be sent by means of a generic transmission system which intrinsically produces transmission errors. The data/information block is made up of a certain number *h* (equal, in this specific case, to 16) of strings of symbols  $(I_{1,1} \ I_{1,2} \ I_{1,3} \ \dots \ I_{1,7}; \ I_{2,1} \ I_{2,2} \ I_{2,3} \ \dots \ I_{2,7}; \ \dots; \ I_{16,1} \ I_{16,2} \ I_{16,3} \ \dots \ I_{16,7})$  having a length *k*, in this example *k* = 7. The *Check* block with the parity (redundancy) symbols is added to the data block and is made up of a number  $(n - k) \cdot h$  of parity symbols (in the example,  $(15 - 7) \cdot 16$ ). For the sake of simplicity only, without wishing to affect the general scope of the invention, the same number of parity symbols has been chosen, i.e. 4 columns (of 16 rows) are for parity A and 4 columns (of 16 rows) are for parity B. There is nothing to prevent different numbers from being considered for each of the parities A and B.

According to a possible embodiment of this invention (see Figures 3b and 3c which show two possible types of interleaving, without and with an exchange of columns respectively), the data block is interleaved by carrying out a permutation within the single columns, that is to say by exchanging the positions of the elements within each single column. Conveniently, for the sake of simplicity of construction and description, the permutation within the single columns may consist of a rotation of the elements. The rotation is in any case a sub-class of permutations.

In the example, the order of the elements in the first column is not altered ("rotation 0"); in the second column a "rotation by 1" is carried out (the second element is moved to the first row); in the third column a "rotation by 2" is carried out (the third element is moved to the first row); ... in the seventh column a "rotation by 6" is carried out (the seventh element is moved to the first row).

The block of redundancy symbols (with redundancies A and B) is interleaved in the same way: for example, a "rotation by 7" is carried out in the eighth column and a "rotation by 14" is carried out in the fifteenth column.

Fig. 3c shows how it would be possible, if required, to carry out a permutation between columns for the parity *Check* block (the columns of symbols A are exchanged with the columns of symbols B). In any case, depending on the column in which the parity symbols are situated, a corresponding rotation is carried out. For the data, on the other hand, permutation between columns is not carried out since it would not bring about any improvement in the performance of the code.

It is obvious that the data block that is actually transmitted is the one shown in Fig. 3a. The block shown in Figures 3b or 3c illustrates clearly that the same data block is read in a different manner by code words. In other words, according to this invention, the code

words consist of first horizontal sequences of symbols of the block [Inf + Check], for example  $I1,1 \ I1,2 \ I1,3 \ \dots \ B1,4$  or  $I6,1 \ I6,2 \ I6,3 \ \dots \ B6,4$  and of second horizontal sequences of symbols of the interleaved block [Inf + Check], for example  $I1,1 \ I2,2 \ I3,3 \ \dots \ B15,4$  or  $I5,1 \ I6,2 \ I7,3 \ \dots \ B3,4$  (if Fig. 3b is considered), or  $I1,1 \ I2,2 \ I3,3 \ \dots \ A15,4$  or  $I5,1 \ I6,2 \ I7,3 \ \dots \ A3,4$  (if the same sequences are considered with reference to Fig. 3c). These same second sequences of symbols can also obviously be read in Fig. 3a by following a diagonal path and "moving upwards" in the column subsequent to the one of the element at the lower end of the diagonal itself.

The code words are paths. A "path" is therefore understood to be a sequence of positions of symbols. Path " $i$ " is given by the sequence *row*  $f(i, j)$  , *column*  $j$  , with  $j = 1,2,3,\dots,n$ . Each of the  $h$  rows is a code word having a length  $n$  and a redundancy  $(n - k)/2$ ; each of the  $h$  paths is also a code word having a length  $n$  and a redundancy  $(n - k)/2$ .

The interleaver according to this invention defines the permutations (which can, in particular, be rotations) and it therefore defines the trajectories. Clearly, the function  $f(\cdot)$  defines the structure of the interleaver. For example, paths along the diagonals are obtained with the following formula:

$$f(i, j) = (i - 1 + j - 1) \bmod h + 1$$

with  $j = 1,2,3,\dots,n$  and  $i = 1,2,3,\dots,h$

It will therefore be understood that this "super code" is systematic (it lets the  $k \cdot h$  information symbols to pass through unaltered and simply adds the  $(n - k) \cdot h$  parity symbols).

This invention, advantageously, achieves the object of equalising the robustness of all the symbols of the code so that there are no areas more fragile than others, by

introducing the use of the rotating interleaver. The sequence of rotations is, theoretically, generic, provided it keeps the number of intersections to a minimum. It is reiterated that, as a generalisation, the rotations are in fact generic permutations.

If we consider two parities (parity A and parity B), the two parities are constructed in such a way that they can protect each other. In the example, the parities are of the same size, although it is possible to have parities of different sizes. Parity B is calculated by rows, having as its input one row of information symbols and one de-interleaved row of Parity A (Fig. 3a). *Vice versa*, in Fig. 3c, parity A is also calculated by rows, but having as the input one interleaved row of information symbols and one row of interleaved Parity B. It must be noted that the permutation on columns has been carried out only on the parity and not on the data. As will be seen, to obtain this loop, it is necessary to solve a linear system.

In Fig. 3b, parity B serves both for the code of Fig. 3a and for the code of Fig. 3b with the exception of an interleaving + the addition of redundancy A which enables the code to be constructed.

It must be noted that the process can be extended to more than two concatenated codes, three for example (using two different interleavers simultaneously). Again in this case, for practical reasons only, redundancies A, B, C have the same lengths. This situation is illustrated schematically in Figures 4a, 4b e 4c.

In the illustrated example, parity C (Fig. 4a) is calculated by rows, having as its input one row of information symbols, one row of Parities A interleaved through a second interleaver and one row of Parities B interleaved through a first interleaver. Parity B (fig. 4b) is calculated by rows, having as its input one row of information symbols interleaved through the first interleaver, one row of Parity A interleaved through the first interleaver

and one row of Parity C interleaved through the second interleaver. Lastly, parity A (Fig. 4c) is calculated by rows, having as its input a row of information symbols interleaved through the second interleaver, one row of Parities C interleaved through the first interleaver and one row of parities B interleaved through the second interleaver. It is in any case clear that the interleaver on the parities may differ from the interleaving on the information.

Construction of the self-protected parities of the code (in transmission):

Having dealt extensively with the interleaver, the problem of constructing the self-protected parities will now be considered. In the case of two blocks of parities (two blocks of parities for the sake of simplicity and in order to be consistent with Figures 3a, 3c), it is necessary to find two sets of columns of parities such that both the rows and the paths are code words. Similarly, in the case of  $x$  blocks of parities, the  $x$  sets of columns having parities such that both the rows and the paths are code words must be found.

Since the code is linear, we exploit this linearity, that is to say we apply the overlapping of effects. The second group of columns of parities is given by the contribution of the data parities ( $P1$  if the parity is per rows or  $P2$  if the parity is per paths) plus the contribution of the parities of the first group of columns  $Y$  ( $AY$  if the parity is per rows or  $BY$  if the parity is per paths). The  $A$  and  $B$  matrixes depend on the code and on the interleaver and can be calculated by means of known techniques.

Let us now find  $Y$ , that is to say the symbols in the first group of columns, such that

$$P1 + A \cdot Y = P2 + B \cdot Y$$

( $A$  and  $B$  are square matrixes with  $(n - k)/2 \cdot h$  rows and  $(n - k)/2 \cdot h$  columns)

The following results:

$$Y = (A - B)^{-1} \cdot (P2 - P1)$$

Therefore  $Y$  will be the first set of columns and  $P1 + AY$  will be the second set.

Naturally, in order to calculate  $Y$  it shall be possible to reverse the matrix  $(A - B)$  (otherwise the system cannot be solved). In order to make it possible to reverse the matrix  $(A - B)$ , degrees of freedom are available consisting of the choice of the interleaver and the choice of the two codes.

The same principle as above is applicable to various different types of interleaver, including the rectangular interleaver (see, for example, Fig. 2c). Fig. 2c shows a “shortened” PVC in which the two redundancies protect each other mutually. In the same way, this property is obtained by solving a linear system. It will be understood that this aspect is totally new as compared with the situation shown in Fig. 2b and originates directly from this invention.

With reference to Fig. 5, which shows an encoder 10 according to this invention (in this case, a two-dimensional encoder),  $Y$  is calculated in a functional block 18 that acts as a linear combiner. The device shown in Fig. 5 receives an input signal (data) to be transmitted. The signal is organised in frames with  $h$  rows and  $k$  information symbols for each row. The frames are sent to a first Short Block Coding 12 which receives the input streams of  $k$  symbols of consecutive rows and produces a first sequence of  $(n - k)/2$  redundancy symbols ( $P1$  as already defined). Each first sequence of redundancy symbols refers to a corresponding row of the frame.

The frames are also sent to an interleaver 14 in which the streams of  $k$  symbols of consecutive rows are submitted to vertical interleaving (within the single columns) with cyclic vertical shifting of the symbols of a column as compared with the symbols of the previous column (if simple rotation is implemented). Subsequently, the interleaved frames are sent as inputs to a second Short Block Coding 16 that produces a second sequence of

$(n - k)/2$  redundancy symbols ( $P2$ ). Each second sequence of redundancy symbols refers to a corresponding diagonal path in the frame.

The outputs from the first and from the second Short Block Coding 12, 16 are sent as inputs to the linear combiner 18. Using the formulae indicated above, the linear combiner 18 produces a total of  $(n - k) \cdot h$  redundancy symbols which are, indeed, the redundancy part to be added (adder 20) to the data frame proper in order to obtain the desired code.

Generally speaking, if more than two blocks of redundancy are envisaged (three or four, for example), one or two (or more) corresponding additional branches must be envisaged with an interleaver and a Short Block Coding, whose output feeds the linear combiner 18. The additional interleaver may be identical to (but it may also differ from) the interleaver 14.

It should be explained that the linear system referred to above is not solved by the linear combiner, instead, it is calculated beforehand in the design stage.

The process described above is entirely general, regardless of the structure of the interleaver, which implicitly defines the matrices  $A$  and  $B$  and the calculation of  $P1$  and  $P2$ .

In any case, making a suitable choice of interleaver (function  $f(i, j)$ ) can make the calculation of  $Y$  very simple. For example, it is possible to make A and B block diagonals resulting in a localisation of the calculations.

#### Structure of the code (receiving side):

Decoding is an iterative process that has been widely dealt with in literature. It consists, in the simplest of cases, of applying alternately the decoder for rows and the decoder for paths a certain number of times. If the number of errors introduced by the

channel does not exceed the correcting capacity of the code, the word that was transmitted is reconstructed perfectly.

In any case, from the point of view of the decoder, this code has the same properties as a Product Vector Code. Only the paths on which the code words are found change. Therefore, correction algorithms applied to the Product Vector Code, which are known in literature, can conveniently be used.

There have thus been shown and described a novel method and a novel device for implementing a linear block code which fulfil all the objects and advantages sought therefor. Many changes, modifications, variations and other uses and applications of the subject invention will, however, become apparent to those skilled in the art after considering the specification and the accompanying drawings which disclose preferred embodiments thereof. All such changes, modifications, variations and other uses and applications which do not depart from the spirit and scope of the invention are deemed to be covered by the invention which is limited only by the claims which follow.